

PATENT ABSTRACTS OF JAPAN

REFERENCE 3

(11)Publication number : 04-186447

(43)Date of publication of application : 03.07.1992

(51)Int.Cl. G06F 12/00
G11B 20/12
G11B 27/00

(21)Application number : 02-316480

(71)Applicant : CANON INC

(22)Date of filing : 21.11.1990

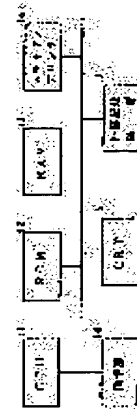
(72)Inventor : OKUDA OSAMU

(54) DIRECTORY MANAGEMENT SYSTEM

(57)Abstract:

PURPOSE: To shorten a retrieving time without reducing effective recording capacity by reading out respective directory entires recorded in an information recording medium and converting respective directory entries into hierarchical structure based upon their attributes and master information.

CONSTITUTION: This directory management system has an external storage device 17 for filing various data in the information recording medium. Respective directory entires for managing respective files are individually recorded in the medium and these entries are read out and converted into hierarchical structure to form a virtual directory in a prescribed area of a RAM 13 and manage the files by the hierarchical structure. Thereby, it is unnecessary to uselessly rerecord also directory entries not to be changed at the time of adding or changing a file, recording capacity can be saved and the retrieving time can be shortened.



⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A) 平4-186447

⑮ Int. Cl.³

識別記号

庁内整理番号

⑬ 公開 平成4年(1992)7月3日

G 06 F 12/00
G 11 B 20/12
27/005 2 0 J
B8944-5B
9074-5D
8224-5D

審査請求 未請求 請求項の数 1 (全8頁)

⑭ 発明の名称 ディレクトリ管理方式

⑰ 特 願 平2-316480

⑱ 出 願 平2(1990)11月21日

⑲ 発 明 者 奥 田 治 神奈川県川崎市中原区今井上町53番地 キヤノン株式会社
小杉事業所内

⑳ 出 願 人 キヤノン株式会社 東京都大田区下丸子3丁目30番2号

㉑ 代 理 人 弁理士 川久保 新一

明 細 書

1. 発明の名称

ディレクトリ管理方式

2. 特許請求の範囲

情報記録媒体にディレクトリエントリを個別に記録するとともに、各ディレクトリエントリの情報として、その属性と親ディレクトリを識別する親情報を含む情報を記録する記録手段と；

情報記録媒体に記録された各ディレクトリエントリを読み出して、その属性と親情報から、各ディレクトリエントリを階層構造に変換する変換手段と；

を有することを特徴とするディレクトリ管理方式。

3. 発明の詳細な説明

[産業上の利用分野]

本発明は、特に記録データの書き換えが不可能

な情報記録媒体に記録されたファイルを階層ディレクトリ管理するディレクトリ管理方式に関する。

[従来の技術]

従来より、情報記録媒体に記録したデータを管理する手法としてディレクトリ管理方式が知られている。

このディレクトリ管理方式は、データのブロックをファイルとし、また、1つのファイルを管理するための情報をディレクトリエントリとし、さらに、まとめて管理したいファイルのディレクトリエントリをまとめたディレクトリエントリ群をディレクトリとして、ディレクトリによって情報記録媒体に記録したファイルを管理するものである。

また、磁気ディスク等の記録データの書き換えが可能な情報記録媒体においては、ファイルを階層構造で管理するためのディレクトリ管理方式が、従来より一般的に行われている。

以下、磁気ディスクでの階層ディレクトリ管理

特開平4-186447 (2)

方式の一例を説明する。

この磁気ディスクの階層ディレクトリ管理方式では、ディレクトリは通常のファイルと同様に扱いディレクトリ単位でファイルを管理する。

そして、最上位のディレクトリをルートディレクトリといい、その他の下位のディレクトリはサブディレクトリといい、さらにサブディレクトリの上位ディレクトリを親ディレクトリという。

第8図は、磁気ディスクにおけるディレクトリエントリフォーマットを例示する模式図である。

図示のように、ファイル情報として、ファイル名501、属性502、位置情報503、ファイルサイズ504を含んでいる。

このうちファイル名501は、ユーザがファイルを識別するために付ける名称である。

また、属性502は、ディレクトリエントリが示す情報が通常ファイルなのかサブディレクトリなのかを識別するための情報で、属性がサブディレクトリのとき、そのディレクトリエントリが示

クトリを示している。

ルートディレクトリには、ファイルF1とサブディレクトリAのディレクトリエントリが記録される。

サブディレクトリAには、ファイルF2およびサブディレクトリBのディレクトリエントリと、親ディレクトリであるルートディレクトリとサブディレクトリA自身のディレクトリエントリとが記録される。

サブディレクトリBには、ファイルF3およびファイルF4のディレクトリエントリとともに、親ディレクトリであるサブディレクトリAとサブディレクトリB自身のディレクトリエントリが記録される。

ルートディレクトリからサブディレクトリAに移るときは、ルートディレクトリ中のサブディレクトリAのディレクトリエントリによって、記録媒体からサブディレクトリAを読み出して、サブディレクトリAに含まれるディレクトリエントリによってディレクトリAが管理するファイルにア

すのはディレクトリエントリ群すなわちディレクトリである。

さらに、位置情報503は、ファイルが磁気ディスクのどの場所から記録されているかを示す情報であり、ファイルサイズ504は、ファイルの有効なバイト数を示している。

また、上記サブディレクトリには、親ディレクトリとサブディレクトリ自身のディレクトリエントリを含めて記録する。これにより、ディレクトリからディレクトリへのアクセスを可能にし、階層ディレクトリ構造を実現している。

そして、この管理方式では、ディレクトリ単位でファイルを管理し、ディレクトリを移る場合は、その移るディレクトリのディレクトリエントリにより情報記録媒体からディレクトリを読み出してファイルにアクセスする。

第3図は、階層ディレクトリ構造のモデル例を示す模式図である。

図において、R00はルートディレクトリ、F1～F4は通常ファイル、A、Bはサブディレ

クセスする。

また、サブディレクトリAからルートディレクトリに移るときは、サブディレクトリA中のルートディレクトリのディレクトリエントリにより、記録媒体からルートディレクトリを読み出してルートディレクトリに含まれるディレクトリエントリによってルートディレクトリが管理するファイルにアクセスする。

ファイルの追加、更新によってディレクトリに変更が生じた場合、例えば、サブディレクトリAにファイルF5を新たに追加した場合には、サブディレクトリAにファイルF5のディレクトリエントリを追加してサブディレクトリAを全て再記録する。

また、ファイルF5の内容を変更した場合も、サブディレクトリA中のファイルF5のディレクトリエントリを変更し、サブディレクトリAを全て再記録する。

以上の説明のように、磁気ディスクにおける階層ディレクトリ管理方式ではディレクトリをファ

特開平4-186447 (8)

イルと同様に扱い、一つでもディレクトリエントリに変更が生じたときには、変更したディレクトリエントリを含むサブディレクトリまたはルートディレクトリの情報を全て再記録する。

〔発明が解決しようとする課題〕

しかしながら、記録データの消去が不可能な情報記録媒体で磁気ディスクと同様の階層ディレクトリ管理方式を行うと、ファイルの追加、変更の度にサブディレクトリまたはルートディレクトリを新たに追記しなければならず、変更する必要の無いディレクトリエントリまで再記録することから、変更のない多くのディレクトリエントリを重複して記録することになり、これらが無駄に残存したままとなり、有効な記録容量を減らすうえ、ディレクトリの検索時間も長くなる欠点がある。

本発明は、記録データの書き換えが不可能な情報記録媒体に対する階層ディレクトリ管理に適したディレクトリ管理方式を提供することを目的とする。

第1図は、本発明の一実施例によるディレクトリ管理方式を用いた情報処理装置の構成を示すブロック図である。

この情報処理装置は、この実施例における各種処理を実行するCPU11と、このCPU11の制御プログラム等を格納したROM12と、CPU11が制御動作を実行するためのワークエリアとして用いられるRAM13と、オペレータが各種データの入力を行なう操作部14と、オペレータに対する各種表示を行なうCRT15と、画像情報の読み取りや出力を行なうスキャナ/プリンタ部16と、光ディスクやハードディスク等の情報記録媒体に各種データをファイルする外部記録装置17とを有している。

そして、この実施例では、上記情報記録媒体には、各ファイルを管理するためのディレクトリエントリを個々に記録するとともに、これらのディレクトリエントリを読み出して階層構造に変換することにより、上記RAM13の所定領域に仮想的なディレクトリを作りファイルを階層構造で管

〔課題を解決する手段〕

本発明は、情報記録媒体にディレクトリエントリを個別に記録するとともに、各ディレクトリエントリの情報として、その属性と親ディレクトリを識別する親情報を含む情報を記録する記録手段と、情報記録媒体に記録された各ディレクトリエントリを読み出して、その属性と親情報から、各ディレクトリエントリを階層構造に変換する変換手段とを有することを特徴とする。

〔作用〕

本発明では、ディレクトリを情報記録媒体に記録するのではなく、個々のディレクトリエントリを情報記録媒体に記録するとともに、記録した各ディレクトリエントリを読み出して、各ディレクトリエントリを階層構造に変換することにより、仮想的なディレクトリを作りファイルを階層構造で管理することにより、ファイルの追加、変更の際、変更のないディレクトリエントリまで無駄に再記録する必要を無くすることができる。

〔実施例〕

理するものである。

第2図は、上記ディレクトリエントリのフォーマット例を示す模式図である。

シリアル番号101は、各ディレクトリエントリを識別する情報である。

ファイル名102は、ディレクトリエントリが示すファイルに付ける名称である。

属性103は、ディレクトリエントリの種類を判別するものである。

位置情報104は、ディレクトリエントリが管理するファイルが記録媒体のどの場所（セクタアドレス等）に記録されているかを示すものである。

ファイルサイズ105は、ディレクトリエントリが管理するファイルの大きさ（使用セクタ数、バイト数等）を示すものである。

親情報106は、ディレクトリエントリの親ディレクトリを識別する情報である。

なお、これらの情報はどのような順序で記録されていてもかまわない。

特開平4-186447 (4)

第3図は、階層ディレクトリ構造のモデルを例示する模式図である。

図において、Rootはルートディレクトリ、F1～F4は通常ファイル、A、Bはサブディレクトリを示している。

本実施例では、メモリ(RAM13)内に設けた仮想ディレクトリ展開領域上に、第3図に示すような階層構造のディレクトリを展開できるように、各ディレクトリエントリの各情報を適宜設定する。

第4図は、情報記録媒体に記録された各ディレクトリエントリの一覧表を示す模式図である。

記録媒体上には、サブディレクトリA、BのディレクトリエントリとファイルF1～F4のディレクトリエントリおよびファイルF1～F4が第2図のフォーマットで記録されている。

各シリアル番号は、記録したディレクトリエントリに、その記録順に付け、他のディレクトリエントリの親情報として用いる。ルートディレクトリは情報記録媒体には記録しないが、シリアル番

号は0とする。したがって、ルートディレクトリのシリアル番号0はファイルF1とサブディレクトリAの親情報としてのみ記録されている。

また、属性は、通常ファイルとサブディレクトリがあり、fが通常ファイル、sがサブディレクトリを示している。属性がサブディレクトリのディレクトリエントリは、記録媒体上のデータを管理するのではなく、データを読み出して階層構造にしたときのサブディレクトリの名前(ここではAまたはB)を表すものである。したがって、ディレクトリエントリの集合としてのサブディレクトリA、B自体は、記録媒体上には記録されていない。

さらに、位置情報のF1-TOP、F2-TOP、F3-TOP、F4-TOPは、それぞれファイルF1、F2、F3、F4が記録媒体上のどこから記録されているかを示し、ファイルサイズのF1-SIZE、F2-SIZE、F3-SIZE、F4-SIZEは、それぞれファイルF1、F2、F3、F4のバイト数を示す。

位置情報とファイルサイズは、属性が通常ファイルの時にだけ有効で、属性がサブディレクトリするときには意味を持たない。

各ディレクトリエントリは、情報記録媒体に1つずつ書き込み、読み出しができるようになっている。また、ディレクトリエントリの読み出しはシリアル番号の1から順に読み出す。

なお、以後の説明では、サブディレクトリA、Bのディレクトリエントリを、ディレクトリA、Bとし、ファイルF1～F4のディレクトリエントリをディレクトリF1～F4という。

第5図は、第4図に示すディレクトリエントリデータを読み出し、メモリ(RAM13)内の仮想ディレクトリ展開領域上で階層構造に展開した場合のディレクトリエントリデータを示す模式図である。

第5図において、Rootは、ルートディレクトリのディレクトリエントリデータの先頭を示し、subA、subBは、それぞれサブディレクトリA、サブディレクトリBのディレクトリエ

ントリデータの先頭を示す。

また、nextはルートディレクトリまたはサブディレクトリに属すディレクトリエントリをリンクしていくためのもので、リンクした次のディレクトリエントリのアドレスを示す。

さらに、nullは、リンクしたディレクトリエントリが最後であることを示す。

また、forward、backは、属性がサブディレクトリのときだけセットする情報で、forwardは、親ディレクトリの先頭アドレスを示し、backはサブディレクトリに属す最初のディレクトリエントリのアドレスを示す。

第6図は、第4図に示すディレクトリエントリデータを、第5図に示すようにメモリ上に展開していく際に使うディレクトリテーブルを示す模式図である。このディレクトリテーブルは、RAM13の上記第5図に示す仮想ディレクトリ展開領域とは別の領域に設けられている。

このディレクトリテーブルには、ルートディレ

特開平4-186447 (5)

クトリおよびサブディレクトリのシリアル番号と先頭アドレスとを記憶するものであり、ディレクトリエントリデータを展開する際、その親情報に対応するディレクトリの先頭アドレスを示し、仮想ディレクトリ展開領域における検索動作を行なわせる機能をもつ。

また、1つもディレクトリエントリを読み出してない初期状態では、第6図のディレクトリテーブルには何も登録されていないことを示すnullをセットする。

なお、第5図に示す仮想ディレクトリ展開領域におけるRootのアドレスと、第6図に示すディレクトリテーブルの先頭アドレスとは、予め決められているものとする。

第7図は、ディレクトリエントリを階層構造に変換する処理の概要を示すフローチャートである。

まず、ディレクトリエントリを情報記憶媒体から読み出し、ディレクトリエントリの親情報を読み取る(S1)。なお、ディレクトリエントリの

この後、そのディレクトリエントリの属性を判別し、属性がサブディレクトリでなければS8に進み、サブディレクトリならば、サブディレクトリの親ディレクトリへの飛び先をディレクトリテーブルに記憶し(S7-1)、さらに、ディレクトリテーブルにサブディレクトリのディレクトリエントリのシリアル番号と飛び先を記憶して(S7-2)、S8に進む。

次に、全ディレクトリエントリを読み出したかどうかを判別し(S8)、全ディレクトリエントリを読み出すまで、S1から繰り返す。

以上のようにして、記憶媒体内の各ディレクトリエントリを、RAM13内の仮想ディレクトリ展開領域に順次登録していき、階層構造のディレクトリを作成する。

次に、このような変換処理を、第4図のディレクトリエントリデータの内容に即して詳細に説明する。

まず、シリアル番号1のディレクトリF1を読み出す。ディレクトリF1は、属性が通常ファイ

読み出しは、シリアル番号順に行なう。

次に、予め設定されているアドレスに基づいてディレクトリテーブルを参照し、このテーブルのシリアル番号にS1で読み取った親情報があるかどうか検索し(S2)、ディレクトリテーブルにシリアル番号がない場合には、最初のディレクトリエントリであると判断し(S3)、ディレクトリテーブルに、予め決められているルートディレクトリの飛び先とシリアル番号0とを記憶する(S3-1)。

次に、上記親情報によってディレクトリテーブルを検索し、この検索結果に基づいて上記第5図の仮想ディレクトリ展開領域における各ディレクトリの先頭アドレスに飛び(S4)。そして、ここに既に登録されているディレクトリで検索を続け、次のディレクトリエントリへの飛び先を示す情報を求めていき(S5)、最後のディレクトリエントリを示すnullを検出した時点で、ディレクトリエントリを仮想ディレクトリ展開領域に登録する(S6)。

ルで親情報が0である。そこで、第6図のディレクトリテーブルからシリアル番号0を検索する。

しかし、ここではディレクトリテーブルには、nullしか格納されていないので最初のディレクトリだと判断し、ディレクトリテーブルにシリアル番号0、ルートディレクトリの先頭を示すRootを記憶する。

そして、仮想ディレクトリ展開領域のRootにディレクトリF1を登録する。このとき次のディレクトリエントリへの飛び先を示すnextにはnullをセットする。

次に、全ディレクトリエントリを読み出してないので、シリアル番号2のディレクトリAを読み出す。ディレクトリAは属性がサブディレクトリで親情報が0である。

そこで、ディレクトリテーブルからシリアル番号0を検索し、シリアル番号0の飛び先である仮想ディレクトリ展開領域のRootに行く。ここで、Rootのディレクトリエントリのnext

特開平4-186447 (6)

がnullとなるまで検索する。nextがnullとなったところで、このnextに、次に登録するディレクトリエントリの仮想ディレクトリ展開領域上のアドレスをセットし、このアドレスでディレクトリAを登録する。

このときディレクトリAの属性がサブディレクトリなので親ディレクトリにアクセスするためにforwardに親ディレクトリであるルートディレクトリの先頭を示すRootをセットし、サブディレクトリAに属するディレクトリエントリにアクセスするためのnextには何も登録しないことを示すnullをセットし、ディレクトリテーブルにディレクトリAを登録したアドレスSubAとシリアル番号2を記憶する。

次に、シリアル番号3のディレクトリF2を読み出す。ディレクトリF3は属性が通常ファイルで親情報が2である。

そこで、ディレクトリテーブルからシリアル番号2を検索する。そして、ディレクトリテーブルからシリアル番号2の飛び先であるSubAに行

く。ディレクトリAに属するディレクトリエントリへの飛び先を示すnextにはnullがセットされているのでサブディレクトリAに属するディレクトリエントリが無いと判断する。そして、ディレクトリAのbackにディレクトリF2を登録する仮想ディレクトリ展開領域上のアドレスをセットし、このアドレスにディレクトリF2を登録する。

このように記憶して全てのディレクトリエントリを登録した結果、第5図に示す階層構造を得ることができる。

なお、本実施例ではディレクトリエントリを1つずつ記憶媒体から読み出しているが全てのディレクトリエントリを読み出した後、階層構造に変換しても良い。

第5図のように展開したディレクトリによって、第3図に示す階層構造でファイルを管理することができる。

すなわち、ルートディレクトリからサブディレクトリAに移るときは、ルートディレクトリRo

otからディレクトリエントリを検索してサブディレクトリAのディレクトリAが管理するファイルにアクセスする。

また、サブディレクトリAからルートディレクトリに移るときには、サブディレクトリAのディレクトリエントリのforwardによってルートディレクトリの先頭であるRootに飛ぶことによりルートディレクトリが管理するファイルへのアクセスも実現できる。

また、ファイルを追加、変更する場合には、例えば、サブディレクトリAの下にファイルF5を追加する場合、まず、属性を通常ファイル、親情報をサブディレクトリAのシリアル番号2としてファイルF5のディレクトリエントリのみ追記し、メモリ上のディレクトリAのリストを検索して行きディレクトリBにリンクすれば良い。

また、ファイルF5を変更する場合には、新たにファイルF5のディレクトリエントリを追記して、メモリ上のディレクトリF5の内容を変更すれば良い。

また、後にディレクトリを階層構造に変換する段階で、同じファイル名を検出したときにはシリアル番号の大きな方、つまり後に記憶されたディレクトリを有効とすれば良い。

以上のように本実施例では、全ディレクトリエントリを読み込んでメモリ上に保持しているのでディレクトリを移るとき再び記憶媒体上からディレクトリを読み出す必要がなく、またディレクトリエントリに変更が生じた場合にも、変更するディレクトリエントリだけを記憶するので記憶時間も短縮できる。

なお、上記実施例では、各ディレクトリエントリを識別する情報として、各ディレクトリに付けたシリアル番号を用いたが、ファイル名または各サブディレクトリのみを識別する情報を付与してこれを識別に用いてもよい。また、ディレクトリに含む属性と親情報からディレクトリを階層構造に変換する手段は、前記の実施例で示したものに限らない。

なお、本発明は記憶データの書き換えができな

特開平4-186447 (7)

い記録媒体だけでなく、データの書き換え可能な記録媒体に対しても階層的ディレクトリ管理を適用することができる。

〔発明の効果〕

以上説明したように、本発明によれば、ファイルの追加、変更によってディレクトリエントリに変更が生じた場合も、変更しなければならないディレクトリエントリを再記録するだけでよく、特に記録データの書き換えが不可能な情報記録媒体に対する階層ディレクトリ管理において、記録容量の節約および検索時間の短縮を図ることができる効果がある。

4. 図面の簡単な説明

第1図は、本発明の一実施例によるディレクトリ管理方式を用いた情報処理装置の構成を示すブロック図である。

第2図は、同実施例におけるディレクトリエントリのフォーマット例を示す模式図である。

第3図は、同実施例における階層ディレクトリ

構造のモデルを例示する模式図である。

第4図は、同実施例において情報記録媒体に記録された各ディレクトリエントリデータの一覧表を示す模式図である。

第5図は、同実施例におけるディレクトリエントリをメモリ上で階層構造に展開した状態を示す模式図である。

第6図は、同実施例においてディレクトリエントリを階層構造に展開していく際に使うディレクトリテーブルを示す模式図である。

第7図は、同実施例においてディレクトリエントリを階層構造に変換する処理の概要を示すフローチャートである。

第8図は、磁気ディスクにおいて従来用いられているディレクトリエントリフォーマットを例示する模式図である。

11…CPU、

12…ROM、

13…RAM、

14…操作部、

15…CRT、

16…スキャナ/プリンタ部、

17…外部記録装置、

101…シリアル番号、

102…ファイル名、

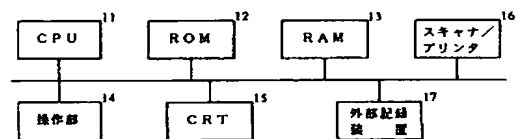
103…属性、

104…位置情報、

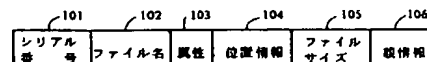
105…ファイルサイズ、

106…親情報。

第1図



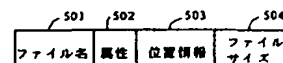
第2図



特許出願人 キヤノン株式会社

代理人 川久保 新一

第8図



特開平4-186447 (8)

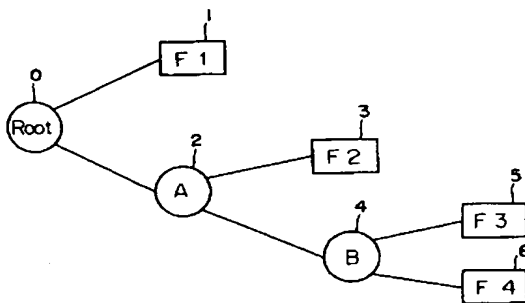
第4図

101 シリアル 番 号	102 ファイル名	103 属性	104 位置情報	105 ファイル サイズ	106 継情報
1	F1	f	F1-TOP	F1-SIZE	0
2	A	s			0
3	F2	f	F2-TOP	F2-SIZE	2
4	B	s			2
5	F3	f	F3-TOP	F3-SIZE	4
6	F4	f	F4-TOP		4

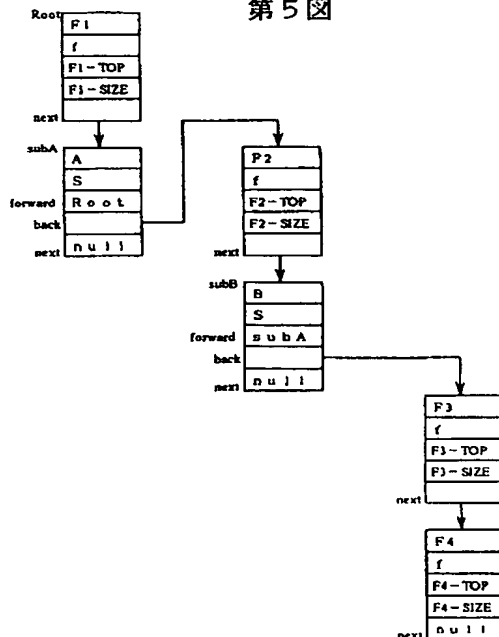
第6図

シリアル 番 号	アドレス
0	Root
2	subA
4	subB
	null

第3図



第5図



第7図

